UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

DATATERN, INC.

Plaintiff,

v.

BLAZENT, INC., MICROSTRATEGY INC., CARL WARREN AND COMPANY INCORPORATED, LANCET SOFTWARE DEVELOPMENT, INC., AIRLINES REPORTING CORP., MAGIC SOFTWARE ENTERPRISES LTD., MAGIC SOFTWARE ENTERPRISES, INC., TERADATA CORPORATION, INFORMATICA CORPORATION, EPICOR SOFTWARE CORPORATION, and PREMIER, INC.

Defendants

1:11-cv-11970-FDS

(Consolidated)

MICROSTRATEGY'S REPLY IN SUPPORT OF MOTION FOR SUMMARY JUDGMENT OF NO INFRINGEMENT

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Page(s)

Cases

I. INTRODUCTION

MicroStrategy's motion turns on a narrow dispositive question: has DataTern identified any MicroStrategy "object model" having object "classes," as mandated by the constructions adopted by the Federal Circuit or stipulated by DataTern? The answer is no, thus entitling MicroStrategy to summary judgment.

Despite the straightforward issue, DataTern fails to identify any "object model" with object classes. Instead, DataTern ignores the applicable claim constructions and cites MicroStrategy's marketing literature out of context. DataTern's opposition ultimately dedicates a single paragraph to the dispositive question. Even in that sole paragraph, DataTern makes only a conclusory contention that the alleged MicroStrategy "object model" contains classes. To buttress this, DataTern ignores its stipulated and binding claim construction of "class," and instead applies an inconsistent and never-before disclosed interpretation of that term. The accused products are fundamentally different from the claimed subject matter, and DataTern has not, and still cannot, articulate a colorable basis for infringement under the binding constructions of "object model" and "class."

DataTern resorts to hyperbole in casting this as a motion for "discovery sanctions." MicroStrategy has not moved for sanctions; it has moved for summary judgment on the merits. Faced with such a motion, DataTern did not articulate a basis for more discovery to fully respond to MicroStrategy's motion, and did not identify additional facts that might affect the outcome. DataTern's theory of infringement, as articulated in its contentions, fails as a matter of law in view of the Federal Circuit opinion construing the claim term "object model," and no amount of additional discovery on a defective theory can salvage it. Both parties have fully briefed that issue on the merits, and it is ripe for summary judgment.

II. ARGUMENT

A. The Federal Circuit's Construction of "Object Model" Requires "Classes".

The Supreme Court has held that "[t]he plain language of Rule 56(c) mandates the entry of summary judgment ... against a party who fails to make a showing sufficient to establish the existence of an element essential to that party's case, and on which that party will bear the burden of proof at trial." *Celotex Corp. v. Catrett*, 477 U.S. 317, 322 (1986). MicroStrategy moved for summary judgment based on DataTern's inability to satisfy the "object model" limitation as construed by the Federal Circuit.

Instead of meeting the substance of MicroStrategy's motion, DataTern relies on a red herring by spending most of its brief's technical discussion on the appearance of the words "object" and "object model" in MicroStrategy's marketing literature. (Opp. at 4–8, 13–16, 20.) But these terms, used in the context of marketing materials, are broad and subject to multiple meanings. Business concepts analyzed with MicroStrategy software are indeed "objects" in the broadest sense, as are trees and cell phones. Even in the context of computer science, the word "object" is a broad term with multiple meanings, as DataTern's former technical expert recognized.[1] Gupta Decl. at 3 n.1 ("[C]omputer programmers often use the word 'object' to generically refer to a structural representation of a concept or entity. As I describe herein, certain contextual uses of the word 'object' have a more precise (and often different) meaning in the art.") (attached as Ex. A). Given this diversity of meanings, MicroStrategy's use of the words "object" and "object model" in its marketing literature to explain its platform is irrelevant, unless they are the specific kinds of "objects" or "object models" at issue in the '502 patent. *Fenner*

[1] In other instances, DataTern acknowledges that MicroStrategy uses terms that happen to appear in the patent but with a different meaning. [Opp. at 5. ("Notably, MicroStrategy does not use the word 'Attribute' in the same sense as the '502 patent.").]

Invs., Ltd. v. Cellco P'ship, 778 F.3d 1320, 1322 (Fed. Cir. 2015) ("The terms used in patent claims are not construed in the abstract, but in the context in which the term was presented and used by the patentee….").

In the context that matters—that of the '502 patent—the Federal Circuit and the Southern District of New York have adopted, over DataTern's objections, a specific construction of the term "object model" that requires "classes." (Mot. at 4–5, 6–7.) As the Federal Circuit held, "[w]e agree with SAP that the district court properly construed 'object model' ***to require classes***." *Microsoft Corp. v. DataTern, Inc.*, 755 F.3d 899, 908 (Fed. Cir. 2014) (emphasis added). In light of this binding construction, DataTern cannot meet its burden of proof by pointing to the use of "objects" and "object model" in MicroStrategy's marketing literature. DataTern must go further and present admissible evidence that MicroStrategy's products provide an "object model" ***with classes***.[2]

B. DataTern Cannot Identify Any "Object Model" with "Classes" in MicroStrategy's Products.

Although it bears the burden of proof, DataTern has not presented any relevant evidence—either in its infringement contentions or even in its opposition papers—to show that MicroStrategy's products have an object model *with classes*. DataTern only devotes a single paragraph to this central issue, in which it contends that MicroStrategy's "object definition"

[2] To be clear, MicroStrategy's products do not "select an object model" for reasons separate and apart from the lack of "object classes." For example, the New York court's construction also required that the claimed "object model" be a template with (1) "a predetermined standardized structure" (2) that "relat[es] to an object-oriented software application" and (3) "includ[es]…inheritance relationships among classes." ***None*** of these three elements of the first claim limitation are satisfied. Moreover, other claim elements are also absent from MicroStrategy's products. Because this motion seeks judgment on DataTern's contentions, it focuses on a single simple and dispositive element that the Federal Circuit previously addressed—the lack of anything corresponding to "classes" in DataTern's contentions.

somehow constitutes a "class." (Opp. at 15.) In other words, unable to find the term in MicroStrategy's documents, DataTern must imply the existence of a "class." Opp. at 16 (arguing that "a class by any other name is still a class").

This argument is misleading, because the "object definitions" on which DataTern relies are ***business data modelling*** concepts, whereas the "classes" required by the claims are ***software programming*** constructs that specifically relate to object-oriented programming.[3] This distinction is most apparent in considering two specific requirements of "class" that are absent from MicroStrategy's "object definitions": a "class" must (i) relate to object-oriented software programming; and (ii) instantiate objects.

1. MicroStrategy's "Object Definitions" Do Not Relate to Object-Oriented Software Programming.

"Class," in the context of the patent, is a term that relates specifically to object-oriented programming. As the New York court explained, "[o]bject-oriented programming typically organizes a software application into a collection of 'objects' that are defined by 'classes.'" *Microsoft Corp. v. DataTern, Inc.*, 2012 WL 3682915 at *6 (S.D.N.Y. Aug. 24, 2012) ("*Microsoft I*") (internal quotation marks and alterations omitted); *see also DataTern, Inc. v.*

[3] The conceptual difference between business data modelling and object-oriented programming is discussed in MicroStrategy's opening brief at page 13. Many illustrations of business data modelling appear in MicroStrategy's documentation, *see generally* ECF No. 56-16, -17, -18. For example, the first step of building a "logical data model" in the MicroStrategy platform involves "identifying the facts," for example "in retail models, sales facts are often stored at the store, item, or day level, meaning that a sale takes place in a particular store, for a particular item, on a particular day." (ECF No. 56-17 at 9.) Business data modelling is thus a high-level process of capturing and organizing real-world business arrangements—the user is concerned not with what is going on "under the hood" in the software but how the business actually works. By contrast, the DataTern patent claims relate to a subfield of computer science concerned with "in-memory instances of data structures," i.e. constructs that are internal to software, as laid out in DataTern's former expert's declaration. *See generally* ECF No. 60 (Gupta Decl.) at ¶¶ 8–9.

Epicor Software Corp., __ Fed. App'x. __, 2014 WL 7234037 at *4 (Fed. Cir. Dec. 19, 2014) ("[C]ommon sense dictates that at some point, code must be generated for the DPerson class…").

This issue was central to the claim construction dispute in the New York case. During the *Markman* process in that case, the parties disputed whether "the correct construction of 'object model' must tie that phrase to the 'object-oriented software application.'" *Microsoft I*, 2012 WL 3682915 at *6. DataTern argued that the term was not so limited, and could instead be *any* "template with a predetermined standardized structure."[4] *Id.* at *4. The New York court disagreed. *Id.* at *6. Instead, it held that "the invention ***requires*** an object-oriented software application," and that "the context of the patent itself limits the standardized template [of the "object model"] to one ***involving object-oriented software applications***." *Id.* (emphasis added). Its construction of "object model" reflected this understanding by defining the term as "a template with a predetermined standardized structure both ***relating to an object-oriented software application*** and including object classes and inheritance relationships among classes." *Id.* In short, "classes" are programming constructs that specifically "relat[e] to an object-oriented software application." *Id.*

In contrast, the "object definitions" in MicroStrategy's literature are *not* programming constructs at all. As DataTern admits in its brief, they are business data modelling concepts: "This object model … is also called the 'Logical Data Model,' a 'Logical Business Model' or just a 'data model' …." (Opp. at 4.) These "object definitions," as used in MicroStrategy's literature, represent a user's understanding of how his business data is arranged, related, and experienced by the general data user or analyst. (SOF ¶ 54–57.) DataTern has thus failed to

[4] DataTern's proposed construction also included additional, non-limiting language that is not at issue here. *Id.* at *4.

demonstrate that these "object definitions" are programming constructs or that they relate to the structure of an object-oriented software application. (Opp. at 15–16.) And DataTern cannot make such a showing because MicroStrategy's "object definition" is conceptually and physically distinct from the programming construct "class." It is not, as DataTern would have this Court believe, "a class by any other name."

2. MicroStrategy's "Object Definitions" Do Not Instantiate by Serving as a Template for Objects.

The "object definitions" on which DataTern relies are not "classes" because they are not templates from which objects can be instantiated, i.e., created.

As the New York court explained: "[C]lasses" are analogous to "a template or cookie cutter from which individual objects are stamped out." *Microsoft I*, 2012 WL 3682915 at *4. This understanding is so well established that DataTern even stipulated to a construction of "class" that includes this instantiation requirement. *Id.* at *4. DataTern's former expert also agreed that "[t]his construction is consistent with how a person of ordinary skill would define the word ['class'] as used in the context of object-oriented programming."[5] Ex. A at ¶ 12.

In its opposition to MicroStrategy's motion, however, DataTern jettisons this instantiation requirement and advances a new definition of "class" without this constraint:

DataTern's new proposed construction for "class"	**Stipulated Construction for "class" that DataTern "is bound by" per the Federal Circuit**
"a definition or template of an object, which includes attributes, and which can also have other characteristics, such as inheritance relationships or methods." (Opp. at 15.)	"a definition that specifies attributes and behavior of objects, and ***from which objects can be instantiated***." (Mot., Ex. T at 2 (emphasis added).)

[5] There is a typographical error in Dr. Gupta's declaration that substitutes the word "object" for the word "class" in the quoted sentence. But the intended meaning is clear in the context.

It is, however, too late for DataTern to abandon its stipulated definition of "class." As the Federal Circuit ruled, "DataTern … is bound by virtue of its stipulation to … [this] construction." *Microsoft Corp. v. DataTern, Inc.*, 755 F.3d 899, 909 (Fed. Cir. 2014). Under this binding definition, there can be no infringement. Not surprisingly, DataTern's opposition makes no attempt to show that objects can be "stamped out" from MicroStrategy's "object definitions." Because it omits a key element of the construction, DataTern's infringement argument must fail.

DataTern's assertion that MicroStrategy's object model "is used to 'define and construct objects that represent any business,'" (Opp. at 15), does not address this deficiency. As described above, these ***business data*** concepts are not the object-oriented ***programming*** objects required by the claims. This distinction is apparent from the very sentence that DataTern quotes from MicroStrategy's literature. Opp. Ex. B at 17 ("if an object changes, every other object dependent on it also changes. This ensures *consistency across business definitions*."). Surrounding passages on the same page likewise confirm that the MicroStrategy "objects" represent business concepts and that they are not object oriented programming objects. *E.g., id.* ("The metadata contains the building blocks or objects necessary *to represent an enterprise's business*").

* * * * *

In short, DataTern has no infringement theory or evidence to account for the settled 'object model' construction, which requires an object model with "classes."[6]

[6] DataTern further contends that, "like the '502 patent, the metadata objects have attributes," and that "[a]lthough inherence relationships and methods are not required … those too are also present in MicroStrategy's product." (Opp. at 15.) These contentions fall short of DataTern's burden because neither shows that the alleged "classes" are programming objects or that they can be used to instantiate the "objects" of the claims.

C. DataTern Cannot Delay or Avoid Summary Judgment by Asking for More Discovery.

1. The Issue Is Ripe for Summary Judgment.

Contrary to DataTern's argument, MicroStrategy seeks summary judgment on the merits, not as a "discovery sanction." (Opp. at 16.) The issue is whether DataTern's *theory* of infringement is legally wrong, not merely whether DataTern has provided enough *evidence* in support of that flawed theory. That issue is ripe. Sufficient evidence exists to support MicroStrategy's Motion, and no further evidence can cure DataTern's theory.

This point does not appear to be in dispute. DataTern has not invoked Rule 56(d), which permits a party to respond to a motion for summary judgment by "show[ing] by affidavit or declaration that, for specified reasons, it cannot present facts essential to justify its opposition." (Opp. at 16–20.) Nor has it identified any "specifi[c] reasons" why it needs additional discovery. (*Id.*) And it has not pointed to any specific discovery that would affect the outcome. (*Id.*) Given DataTern's failure to justify any delay or further discovery, the Court should not deny or postpone summary judgment. *See, e.g., Cevasco v. Nat'l R.R. Passenger Corp.*, 606 F. Supp. 2d 401, 419 (S.D.N.Y. 2009) (granting summary judgment motion and denying request for more discovery where "[t]he issues for which the discovery is sought … [were] not relevant to the resolution of [the] motion.").

2. DataTern's Request for Discovery Is Irrelevant, Belated, and Futile.

DataTern's request for broad-ranging discovery (including source code review that it previously opted to forego) is not only irrelevant to this motion, it is also untimely and unduly burdensome. (Opp. at 17–19.) It has been over two years since MicroStrategy offered its source code for inspection, and since the Court instructed DataTern that, if it thought it needed to inspect the code, "the time to do it is now."[7] Having sat on its rights, DataTern cannot use its own delay to avoid summary judgment.

Nor can DataTern rely on the need to create final infringement contentions as the basis for more discovery and to delay summary judgment. Infringement contentions are supposed to provide a framework for discovery, not the other way around. *See AntiCancer, Inc. v. Pfizer, Inc.*, 769 F.3d 1323, 1329 (Fed. Cir. 2014) (stating preliminary infringement contentions "provid[e] a framework for discovery."). Their purpose "is to narrow the focus of discovery," *L.C. Eldridge Sales Co., Ltd. v. Azen Mfg. Pte., Ltd.*, No. 6:11cv599, 2013 WL 7937026 at *6 (E.D. Tex. October 11, 2013), and thus to reduce the burden on the defendant. To permit discovery without a colorable basis for infringement would be a waste of time and resources.

3. It Is Too Late for DataTern to Amend, Yet Again, Its Infringement Contentions.

As a last ditch effort, DataTern seeks to amend its infringement contentions (again) to "provide even more support for its infringement theories" "as more information is produced" in discovery. (Opp. at 19.) This is a *non sequitur*. If DataTern's theory of infringement is legally wrong under binding claim constructions, then further evidentiary "support" for this defective theory will not make a difference. Such discovery would only burden MicroStrategy and its customers, while wasting limited judicial resources.

DataTern's predicament is one of its own making. It could have, and should have, accounted for an adversary's construction. *St. Clair Intellectual Property Consultants, Inc. v. Matsushita Elec. Indus. Co., Ltd.*, 2012 WL 1015993 (D. Del. March 26, 2012) (providing that,

[7] DataTern proffers an alternative interpretation of this Court's instruction, claiming that—contrary to the Court's plain language—it was really limiting its instruction to source code review for purposes of claim construction. (Opp. at 10–11.) MicroStrategy disagrees with this interpretation, but defers to the Court's understanding of its own instruction.

where a party is aware of a foreseeable claim construction—such as the Federal Circuit's construction here—it must be diligent in accounting for that construction). (Mot. at 16–18.)

Faced with this factually applicable opinion, DataTern attempts, without success, to distinguish it. First, as *St. Clair* makes clear, a party must account for a foreseeable construction in infringement contentions, not just—as DataTern contends, (Opp. at 19)—in expert reports. *Id.* at *5 ("When claim construction remains an open issue at the time the parties serve expert reports ***and infringement contentions***, the parties have an obligation to prepare for the fact that the court may adopt the other party's claim construction."). Second, the early stage of fact discovery in this case is irrelevant, because *St. Clair* turned on the foreseeability of the ignored construction, not the amount of discovery undertaken. *Id.* at *5–6.

Beyond *St. Clair,* DataTern has not identified additional discovery that would affect the outcome here, nor has it addressed MicroStrategy's other cases, which also show that DataTern's request to amend its contentions is improper. (*Compare* Mot. at 16-20, *with* Opp. at 19-20.)

III. CONCLUSION

For the reasons set forth in MicroStrategy's opening brief and herein, the Court should grant summary judgment of non-infringement in favor of Defendants.

Dated: May 1, 2015

/s/ Adam J. Kessel
Adam J. Kessel (BBO #661,211)
FISH & RICHARDSON P.C.
One Marina Park Drive
Boston, MA 02110-1878
Tel: (617) 542-5070
Fax: (617) 542-8906
kessel@fr.com

Benjamin K. Thompson (admitted *pro hac vice*)
FISH & RICHARDSON P.C.
1180 Peachtree St. NE
Atlanta, GA 30318
Tel: (404) 724-2844
bthompson@fr.com

Attorneys for Defendant
MICROSTRATEGY INCORPORATED.

CERTIFICATE OF SERVICE

I certify that this REPLY IN SUPPORT OF MOTION FOR SUMMARY JUDGMENT OF NO INFRINGEMENT is being filed through the Court's electronic filing system on May 1, 2015, which serves counsel for other parties who are registered participants as identified on the Notice of Electronic Filing (NEF). Any counsel for other parties who are not registered participants are being served by first class mail on the date of electronic filing.

/s/ Adam J. Kessel
Adam J. Kessel